

PATENT ABSTRACTS OF JAPAN

(11)Publication number : **08-202529**

(43)Date of publication of application : **09.08.1996**

(51)Int.Cl.

G06F 7/00

G06F 12/04

(21)Application number : **07-008059**

(71)Applicant : **CANON INC**

(22)Date of filing : **23.01.1995**

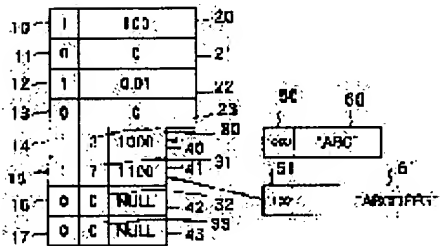
(72)Inventor : **HARA HIROSHI
TAMEGAI MASAHIRO**

(54) METHOD AND DEVICE FOR DATA PROCESSING

(57)Abstract:

PURPOSE: To manage fixed-length data and variable-length data mixedly with ease by transferring flags stored in a storage means and fixed-length data and variable-length data whose flags are effective to a recording medium, and recording them.

CONSTITUTION: The flags 10-17 are provided corresponding to the fixed-length data and variable-length data respectively, and fixed-length data recording areas 20-23 are provided corresponding to those flags 10-17. Further, variable-length data storage areas 60 and 61 as storage means are provided corresponding to those flags 10-17, and the flags 10-17 stored in the variable-length data storage areas 60 and 61 and the fixed-length data and variable-length data whose flags 10-17 are effective are transferred to and recorded on the recording medium by a recording means. Consequently, the fixed-length data and variable-length data can be managed mixedly with ease.



LEGAL STATUS

[Date of request for examination] 28.12.2001

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than withdrawal the examiner's decision of rejection or application converted registration]

[Date of final disposal for application] 19.08.2002

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

*** NOTICES ***

JPO and NCIP are not responsible for any damages caused by the use of this translation.

1. This document has been translated by computer. So the translation may not reflect the original precisely.
2. **** shows the word which can not be translated.
3. In the drawings, any words are not translated.

CLAIMS

[Claim(s)]

[Claim 1] The flag which is the data processor which is intermingled and can process fixed length data and a variable-length data, and was formed corresponding to each of fixed length data and a variable-length data, A storage means to have the fixed-length data storage area prepared by matching with said flag, and the variable-length-data storage region prepared by matching with said flag, The data processor characterized by having a record means by which the flag memorized by said storage means and said flag transmit and record effective fixed length data and an effective variable-length data on a record medium.

[Claim 2] Said variable-length-data storage region is a data processor according to claim 1 characterized by memorizing the address of the memory which has memorized the data length of the variable-length data concerned, and said variable-length data at least.

[Claim 3] Said flag is a data processor according to claim 1 characterized by including the information which shows whether the fixed length data or the variable-length data which corresponds at least is effective, and the information which shows the class of corresponding data.

[Claim 4] The data-processing approach characterized by having the process which will confirm a corresponding flag and will be memorized in memory if it is the data-processing approach of it being intermingled and processing fixed length data and a variable-length data and fixed length data is inputted, and the process which will confirm a corresponding flag and will memorize said variable-length data in memory with the data length and its storing place address of the variable-length data concerned if a variable-length data is inputted.

[Claim 5] The data-processing approach according to claim 4 characterized by having further the process which transmits and records the fixed length data and the variable-length data corresponding to an effective flag on a record medium.

[Claim 6] Said flag is the data-processing approach according to claim 4 characterized by including the information which shows whether the fixed length data or the variable-length data which corresponds at least is effective, and the information which shows the class of corresponding data.

[Claim 7] Said flag is the data-processing approach according to claim 4 characterized by carrying out adjustable [of said fixed-length data storage field] corresponding to said long information including the data length information which shows the data length of corresponding fixed length data further.

DETAILED DESCRIPTION

[Detailed Description of the Invention]

[0001]

[Industrial Application] This invention relates to the data-processing approach of it being intermingled and processing fixed length data and a variable-length data, and its equipment.

[0002]

[Description of the Prior Art] Conventionally, when there are data of 8 bit length, data of 16 bit length, data of 32 bit length, etc. and it treats these data, there are an approach of treating each data as one variable, and the approach of gathering two or more data and treating as one block in the data treated by computer.

[0003] Drawing 9 is drawing showing the example of the DS in the case of treating two or more data as one block, and each of the data C1 with a sign of 8 bit length, the sign-less data C2 of 8 bit length, the sign-less data I1 of 16 bit length, and the variable-length data S1 that treats a character string as one block is shown as fixed length data. When accessing the data in this data block, an identifier is given to the whole data block (the case of drawing 9 A), and it is carried out based on a block name and the identifier added to each data. When accessing as an example the data shown by "C1" of drawing 9, it carries out using notations, such as "A.C1" or "A->C1." Moreover, during the block in drawing 9, when the data block which changed sign-less integer type data into signed-integer mold data is required for coincidence, a new block is created, and another block name (for example, B) is given and managed to the block.

[0004] Therefore, in the above-mentioned conventional example, when many data blocks need to be treated to coincidence, a required number of data blocks must be created, and a name which is different in each block, respectively must be given and managed. Therefore, the need of changing data by modification of a program etc. to all the data blocks to which the modification should be added when a certain data in a data block need to be changed arises. To specifically add the sign-less data of 32 bit length to each of the data block shown by "A" of drawing 10, "B", and "C" newly, it is necessary to add new data to each of three data blocks, a block "A", "B", and "C." Therefore, when need, such as modification of the data in the data block or an addition, arises to many data blocks, modification processing of the data becomes complicated. Moreover, in changing to many data blocks, there is possibility of changing accidentally. For this reason, the DS which prepared a sufficient number of data to the number of data which only sufficient number prepares the data of the longest [data length] in the fixed length data to treat, and is treated also to a variable-length data is used, and how to simplify processing at the time of data modification within a block is also considered.

[0005] Drawing 11 is drawing showing such an example, and 100-109 of a data block 120 show the fixed-length data area. Here, the number of fixed length data is ten pieces. 110-119 in a data block 121 show a variable-length-data field, and the number of these variable-length datas is ten. In this example, access to the data area of each data block can respond by creating two or more data blocks 120 which it is carried out based on the data number expressed with "100+k (0<=k<=9)" or "110+k (0<=k<=9)", and are shown in drawing 11 when two or more data blocks are required, or 121, and both accesses to data control and data can perform it easily.

[0006]

[Problem(s) to be Solved by the Invention] However, in using the DS shown in drawing 11, in order to manage data by making ten fixed length data or ten variable-length datas into a unit, when recording data on an information record medium, it is necessary to record ten fixed length

data or ten variable-length datas. For this reason, when the number of fixed length data or the number of variable-length datas is under "10", respectively, there is a problem that the recording efficiency to an information record medium falls. Moreover, since the die length of data cannot distinguish beforehand at the time of playback of data, in the case of a variable-length data, it is necessary to reproduce data, investigating a data length, and there is a problem that the regeneration efficiency at the time of data playback falls in it.

[0007] This invention was made in view of the above-mentioned conventional example, and aims at offering the data-processing approach that it is intermingled and fixed length data and a variable-length data can be managed easily, and its equipment.

[0008] Moreover, the purpose of this invention is in the data block in which fixed length data and a variable-length data are intermingled to offer the data-processing approach that effective/invalid of each data are judged easily, and can be processed efficiently, and its equipment.

[0009] Moreover, other purposes of this invention are to offer the data-processing approach which records efficiently the data developed in memory on a record medium, and its equipment.

[0010]

[Means for Solving the Problem] In order to attain the above-mentioned purpose, the data processor of this invention is equipped with the following configurations. That is, it has a record means to by_ which the flag memorized by a storage means have the fixed-length data storage area which is the data processor which is intermingled and can process fixed length data and a variable-length data, and was prepared by matching with the flag formed corresponding to each of fixed length data and a variable-length data and said flag, and the variable-length-data storage region prepared by matching with said flag, and said storage means, and said flag transmit and record effective fixed length data and an effective variable-length data on a record medium.

[0011] In order to attain the above-mentioned purpose, the data-processing approach of this invention is equipped with the following configurations. That is, it has the process which will confirm a corresponding flag and will be memorized in memory if it is the data-processing approach of it being intermingled and processing fixed length data and a variable-length data and fixed length data is inputted, and the process which will confirm a corresponding flag and will memorize said variable-length data in memory with the data length and its storing place address of the variable-length data concerned if a variable-length data is inputted.

[0012]

[Function] If the flag which corresponds in the above configuration if fixed length data is inputted is confirmed, and it memorizes in memory and a variable-length data is inputted, a corresponding flag will be confirmed, and it will operate so that the variable-length data may be memorized in memory with the data length and its storing place address of the variable-length data.

[0013]

[Example] Hereafter, the suitable example of this invention is explained to a detail with reference to an accompanying drawing.

[0014] [1st example] drawing 1 is the block diagram showing the outline configuration of the data processor of this example.

[0015] In drawing 1, 130 is CPU and is controlling the whole equipment according to the control program memorized by program memory 131. 132 is RAM, at the time of activation of processing by CPU130, is used as a work area and saves various data temporarily. In addition, this RAM132 is used also for memorizing the data block mentioned later. 133 is the input sections, such as a keyboard and a pointing device, and is used for being operated by the operator and inputting various data. 134 is a hard disk controller (HDC) and is controlling the writing of

the data to a hard disk drive (HDD), and read-out. 137 is displays, such as liquid crystal and CRT, and shows the image data memorized by video memory (VRAM) 138 under control of a display controller (CRTC). 139 is a system bus and has connected each part and CPU130 which were mentioned above.

[0016] Drawing 2 is the conceptual diagram showing the data block memorized by the hard disk 135 of the data processor of this example. Moreover, drawing 3 is the conceptual diagram of the data block in RAM132 of the data processor of this example, and drawing 4 is drawing showing an example of the data stored in the data block shown in drawing 3.

[0017] In drawing 2 - drawing 4, the flag bit (1 bit) which shows whether data were stored in the data areas 20-23 (fixed length data) where 10-13 correspond, and 14-17 are flag bits (1 bit) which show whether the variable-length data is memorized to the corresponding variable-length-data fields 60 and 61. The data area where 20-23 store fixed length data, the data length field (here 8 bits) where 30-33 store the data length of a variable-length data, and 40-43 show the field which has memorized the address of memory with which the variable-length data was stored, respectively. Moreover, 50 and 51 show the address of memory with which the variable-length data is stored, and 60 and 61 show the variable-length data itself memorized by memory.

[0018] In this example, in order to simplify explanation, both the numbers of data of fixed length data and a variable-length data memorized by this data block are made into "4" individuals, and each of the fields 30-33 which show the data length of a variable-length data is made into 8 bit length. Moreover, the class of fixed length data to treat is made into three kinds, 8 bit length, 16 bit length, and 32 bit length, and the data length of fixed length data is explained as 32 bits here.

[0019] When it stores data in the data block of this example, it considers as the condition that store "0" in all the flag bits 10-17 of the data block first shown in drawing 3, and data are stored in neither. Moreover, the "NULL code" with which "0" is stored in the data areas 20-23 which store fixed length data, and the data length fields 30-33, and the 0th street is expressed to the address storage regions 40-43 of a variable-length data is stored, and these data blocks are initialized.

[0020] Next, the data storage processing to the data block shown in drawing 4 is explained.

[0021] It considers as the condition that data were stored in the fixed-length data area 20, by storing "1" in a flag bit 10 first with the application program currently performed by CPU130, when it stores data "100" in the fixed-length data area 20 as data of 8 bit length. Next, "100" which changed the data of 8 bit length into the data of 32 bit length is stored in a data area 20. The data "00000000000000000000 0000001100100" which changed into 32 bit data the data expressed with the binary number "01100100" of 8 bits are stored in this. Like this, when it stores the data "0.01" of 32 bit length in the fixed-length data area 22, after setting "1" to the flag bit 12 corresponding to the fixed-length data area 22 first, the data showing data "0.01" of 32 bit length are stored in a data area 22.

[0022] By the above actuation, as shown in drawing 4, data "100" and "0.01" are stored in each of the fixed-length data areas 20 and 22.

[0023] Next, the case where a variable-length data "ABC" is stored is explained. 8 When it stores a variable-length data "ABC" in the variable-length-data field 60, by storing "1" in a flag bit 14 first, consider as the condition that data are stored in the address storage region 40, and store the die length "3" of a variable-length data "ABC" in the variable-length-data length field 30 with bit data. Next, while determining the memory address of a field 60 which actually memorizes a variable-length data "ABC" and storing a variable-length data "ABC" in the variable-length-data field 60, the address (here "1000" address) of the start address 50 of the data area 60 is stored in the address storage region 40. Like this, in storing a variable-length data "ABCDEFGH", "1" is stored in the flag bit 15 which corresponds first, and it stores in the address storage region 41 the

start-address "1100" address of the variable-length-data field 61 where the die length "7" of this variable-length data "ABCDEFGH" is stored in the field 31 which shows a data length, and this variable-length data "ABCDEFGH" is finally actually stored in it.

[0024] By the above actuation, each of a variable-length data "ABC" and "ABCDEFGH" is stored in RAM132 from each memory address shown in the address storage regions 40 and 41.

[0025] Drawing 5 is the flow chart which shows the processing which memorizes a data block to RAM132 in the data processor of this example, and the control program which performs this processing is memorized by program memory 131, and is performed under control of CPU130.

[0026] It is started by directing the writing of the data to RAM132 by directions [section / under current activation / the application program or the input section 133], and the processing shown with this flow chart writes in and clears "0" to the flag bit and data area of RAM132 at step S1 first. Next, the data which should be progressed and written in step S2 see fixed length data and a variable-length data. If it is a variable-length data, it will progress to step S3, "1" is set to the flag bit to which RAM132 corresponds, it is step S4 and the variable-data length who is going to memorize is memorized to the data length storage region following it. Next, while progressing to step S5, looking for the area of RAM132 which memorizes the variable-length data and memorizing the variable-length data in the area, the start address of the area is memorized to the field to which the address storage regions 30-33 correspond.

[0027] On the other hand, when memorizing fixed length data at step S2, it progresses to step S7, and "1" is set to a corresponding flag bit like the above-mentioned step S3, then it progresses to step S8, and the fixed length data is memorized to the fixed-length data area corresponding to the flag bit. In this way, if it progresses that fixed length data or a variable-length data is memorized by RAM132 to step S6, it judges whether all data were memorized to RAM132 and all are memorized, processing will be ended, but when that is not right, return and the above-mentioned processing are performed to step S2.

[0028] Next, the processing at the time of recording the fixed length data and the variable-length data which were stored in RAM132 by such processing on the information record medium of for example, hard disk 135 grade is explained.

[0029] Thus, in case the data memorized by RAM132 are recorded on HDD135, first, the flag bits 10-17 shown in drawing 4 are recorded, as shown in drawing 2 , and fixed length data 20 and 22 is recorded after this flag bit. Furthermore, data length "3" 30 of a variable-length data "ABC", the information which made the variable-length data "ABC" 1 set, data length "7", and variable-length data "ABCDEFGH" of a variable-length data "ABCDEFGH" are continued and recorded. Therefore, as shown in drawing 2 , "1" is recorded on flags 10, 12, 14, and 15 among flag bits 10-17, fixed length data "100" and "0.01" are recorded following this, and "3", "ABC", "7", and "ABCDEFGH" which are the die length of a variable-length data and the group of a variable-length data further are recorded.

[0030] Drawing 6 is the flow chart which shows the processing which records the data memorized by RAM132 on HDD135, and the control program which performs this processing is memorized by program memory 131.

[0031] It progresses that record to HDD135 is directed to step S10, and all the flag bits are first recorded on HDD135. Next, it progresses to step S11 and the fixed length data corresponding to the flag bit of "1" is recorded on HDD135 one by one. Next, it progresses to step S12, the data length and its variable-length data itself of the variable-length data corresponding to the flag bit which is "1" are made into a group, and it memorizes to HDD135.

[0032] The recording efficiency of the data to a record medium can be gathered by being able to offer the fixed length from whom it is plurality and magnitude differs, and the simple DS which can process variable-length data by single DS by actuation of the above-mentioned example, and

recording only the block with which data are stored on record ***** out of two or more data of a data block.

[0033] The [2nd example] Before storing data in the data block of RAM132, while storing "0" to a flag bit field in the 1st above-mentioned example as initialization of a data block Although initial value is stored also in the fixed-length data areas 20-23 shown in drawing 3 , the data length storage regions 30-33, and the address storage regions 40-43 Since the existence of data [/ based on the value of flag bits 10-17] can be checked, initialization processing can be ended only by storing initial value only in flag bits 10-17. By carrying out like this, the initialization processing to a data storage field can be omitted, and compaction of the processing time can be aimed at.

[0034] In the 1st example of [the 3rd example], when recording the fixed-length data areas 20-23 on HDD135, as shown in drawing 2 , it is recording by assigning a fixed-length field to the fixed-length data areas 20-23. In this case, the field of fixed magnitude must always be assigned to the data length used by CPU130, for example, 8 bit length, 16 bit length, and 32 bit length data. for this reason -- for example, even if the data length of fixed length data is 8 bits, the record section of 32 bit length which is always the maximum data length of fixed length data will be assigned, and it has become the hindrance for gathering recording efficiency. Moreover, in the 1st example, if what kind of thing the data corresponding to a flag bit are cannot distinguish beforehand, when reproducing the data shown in drawing 2 , the break between data cannot be distinguished.

[0035] Then, the above-mentioned trouble is solvable by constituting the flag bit shown in the 1st example from two or more bits, and constituting so that the information which expresses data classification to the bit may be added.

[0036] Drawing 7 is drawing showing the example of the information which can be expressed using this extended flag bit 700, and drawing 8 shows the conceptual diagram of the record condition in the case of recording the data of a data block shown in the 1st example on HDD135 using the flag bit shown in drawing 7 .

[0037] As shown in drawing 7 , a triplet constitutes a flag bit 700 from the 3rd example, and it enables it to express eight kinds of data.

[0038] Moreover, in drawing 8 , the fixed length data "100" shown in the 1st example and "0.01" are treated as data without a sign.

[0039] As shown in drawing 8 , the flag bit which can express the class of data is recorded on fields 70-77 by the flag bit 700 shown in drawing 7 , and the data length can distinguish from the class of each data from flag bits 70 and 71 by it in the fixed-length data areas 20 and 22. That is, in the example of drawing 8 , from a flag bit 70 (101), a data length has no sign at 8 bits, and fixed length data "100" understands [flag bit / 72 (111)] that a data length has no sign at 32 bits for fixed length data "0.01." Furthermore, flag bits 74 and 75 (100) show that a variable-length data is a character string.

[0040] In this way, it can become possible to secure only a required data area to each fixed length data, and the recording efficiency of the data to HDD135 (record medium) can be further raised rather than the 1st above-mentioned example. Moreover, the class of data currently recorded can also be easily distinguished now from this flag bit.

[0041] In addition, even if it applies this invention to the system which consists of two or more devices, it may be applied to the equipment which consists of one device. Moreover, this invention can be applied also when attained by supplying the program which carries out this invention to a system or equipment.

[0042] As explained above, according to this example, in the DS treating two or more data blocks which consist of fixed length data with which two or more data lengths below M differ,

and a variable-length data whose number is a maximum of K, it has two or more fixed length data of data length M, and K variable-length datas for each data block, and a data length can simplify [express / of data / by the data block which has the flag bit which shows an invalid / effective or] the structure of a data block to each data in each data block.

[0043] Moreover, in case a data block is recorded on a record medium, there is effectiveness which can improve the recording efficiency to a record medium by [which record only effective data on a record medium from the contents of the flag bit, and does not record invalid data] constituting like.

[0044]

[Effect of the Invention] As explained above, according to this invention, it is effective in being intermingled and being able to manage fixed length data and a variable-length data easily.

[0045] Moreover, according to this invention, in the data block in which fixed length data and a variable-length data are intermingled, effective/invalid of each data are judged easily, and there is effectiveness which can be processed efficiently.

[0046] Moreover, according to this invention, there is effectiveness which can record efficiently the data developed in memory on a record medium.

TECHNICAL FIELD

[Industrial Application] This invention relates to the data-processing approach of it being intermingled and processing fixed length data and a variable-length data, and its equipment.

PRIOR ART

[Description of the Prior Art] Conventionally, when there are data of 8 bit length, data of 16 bit length, data of 32 bit length, etc. and it treats these data, there are an approach of treating each data as one variable, and the approach of gathering two or more data and treating as one block in the data treated by computer.

[0003] Drawing 9 is drawing showing the example of the DS in the case of treating two or more data as one block, and each of the data C1 with a sign of 8 bit length, the sign-less data C2 of 8 bit length, the sign-less data I1 of 16 bit length, and the variable-length data S1 that treats a character string as one block is shown as fixed length data. When accessing the data in this data block, an identifier is given to the whole data block (the case of drawing 9 A), and it is carried out based on a block name and the identifier added to each data. When accessing as an example the data shown by "C1" of drawing 9, it carries out using notations, such as "A.C1" or "A->C1." Moreover, during the block in drawing 9, when the data block which changed sign-less integer type data into signed-integer mold data is required for coincidence, a new block is created, and another block name (for example, B) is given and managed to the block.

[0004] Therefore, in the above-mentioned conventional example, when many data blocks need to be treated to coincidence, a required number of data blocks must be created, and a name which is different in each block, respectively must be given and managed. Therefore, the need of changing data by modification of a program etc. to all the data blocks to which the modification should be added when a certain data in a data block need to be changed arises. To specifically add the sign-less data of 32 bit length to each of the data block shown by "A" of drawing 10, "B", and "C" newly, it is necessary to add new data to each of three data blocks, a block "A",

"B", and "C." Therefore, when need, such as modification of the data in the data block or an addition, arises to many data blocks, modification processing of the data becomes complicated. Moreover, in changing to many data blocks, there is possibility of changing accidentally. For this reason, the DS which prepared a sufficient number of data to the number of data which only sufficient number prepares the data of the longest [data length] in the fixed length data to treat, and is treated also to a variable-length data is used, and how to simplify processing at the time of data modification within a block is also considered.

[0005] Drawing 11 is drawing showing such an example, and 100-109 of a data block 120 show the fixed-length data area. Here, the number of fixed length data is ten pieces. 110-119 in a data block 121 show a variable-length-data field, and the number of these variable-length datas is ten. In this example, access to the data area of each data block can respond by creating two or more data blocks 120 which it is carried out based on the data number expressed with "100+k (0<=k<=9)" or "110+k (0<=k<=9)", and are shown in drawing 11 when two or more data blocks are required, or 121, and both accesses to data control and data can perform it easily.

EFFECT OF THE INVENTION

[Effect of the Invention] As explained above, according to this invention, it is effective in being intermingled and being able to manage fixed length data and a variable-length data easily.

[0045] Moreover, according to this invention, in the data block in which fixed length data and a variable-length data are intermingled, effective/invalid of each data are judged easily, and there is effectiveness which can be processed efficiently.

[0046] Moreover, according to this invention, there is effectiveness which can record efficiently the data developed in memory on a record medium.

TECHNICAL PROBLEM

[Problem(s) to be Solved by the Invention] However, in using the DS shown in drawing 11 , in order to manage data by making ten fixed length data or ten variable-length datas into a unit, when recording data on an information record medium, it is necessary to record ten fixed length data or ten variable-length datas. For this reason, when the number of fixed length data or the number of variable-length datas is under "10", respectively, there is a problem that the recording efficiency to an information record medium falls. Moreover, since the die length of data cannot distinguish beforehand at the time of playback of data, in the case of a variable-length data, it is necessary to reproduce data, investigating a data length, and there is a problem that the regeneration efficiency at the time of data playback falls in it.

[0007] This invention was made in view of the above-mentioned conventional example, and aims at offering the data-processing approach that it is intermingled and fixed length data and a variable-length data can be managed easily, and its equipment.

[0008] Moreover, the purpose of this invention is in the data block in which fixed length data and a variable-length data are intermingled to offer the data-processing approach that effective/invalid of each data are judged easily, and can be processed efficiently, and its equipment.

[0009] Moreover, other purposes of this invention are to offer the data-processing approach which records efficiently the data developed in memory on a record medium, and its equipment.

MEANS

[Means for Solving the Problem] In order to attain the above-mentioned purpose, the data processor of this invention is equipped with the following configurations. That is, it has a record means to by_ which the flag memorized by a storage means have the fixed-length data storage area which is the data processor which is intermingled and can process fixed length data and a variable-length data, and was prepared by matching with the flag formed corresponding to each of fixed length data and a variable-length data and said flag, and the variable-length-data storage region prepared by matching with said flag, and said storage means, and said flag transmit and record effective fixed length data and an effective variable-length data on a record medium.

[0011] In order to attain the above-mentioned purpose, the data-processing approach of this invention is equipped with the following configurations. That is, it has the process which will confirm a corresponding flag and will be memorized in memory if it is the data-processing approach of it being intermingled and processing fixed length data and a variable-length data and fixed length data is inputted, and the process which will confirm a corresponding flag and will memorize said variable-length data in memory with the data length and its storing place address of the variable-length data concerned if a variable-length data is inputted.

OPERATION

[Function] If the flag which corresponds in the above configuration if fixed length data is inputted is confirmed, and it memorizes in memory and a variable-length data is inputted, a corresponding flag will be confirmed, and it will operate so that the variable-length data may be memorized in memory with the data length and its storing place address of the variable-length data.

EXAMPLE

[Example] Hereafter, the suitable example of this invention is explained to a detail with reference to an accompanying drawing.

DESCRIPTION OF DRAWINGS

[Brief Description of the Drawings]

[Drawing 1] It is the block diagram showing the outline configuration of the data processor of this example.

[Drawing 2] In the data processor of this example, it is the conceptual diagram showing the configuration of the data recorded on a record medium.

[Drawing 3] It is the conceptual diagram showing the DS on RAM in the data processor of this example.

[Drawing 4] It is drawing showing the example of the data storage condition on RAM of the data

processor of this example.

[Drawing 5] It is the flow chart which shows the data write-in processing to the memory in the data processor of this example.

[Drawing 6] It is the flow chart which shows the processing which records the data memorized by the memory in the data processor of this example on HDD.

[Drawing 7] It is drawing showing the condition of the flag bit concerning the 3rd example of this invention.

[Drawing 8] It is the conceptual diagram showing the record condition to the record medium of the data block in the 3rd example of this invention.

[Drawing 9] It is drawing explaining the conventional example of the DS in the case of treating two or more data as a block.

[Drawing 10] It is drawing explaining the conventional example of the DS in the case of treating two or more data as a block.

[Drawing 11] It is drawing explaining the conventional example of the DS in the case of treating two or more data as a block.

[Description of Notations]

10-17 Flag bit

20-23 Fixed-length data area

30-33 Data length field of a variable-length data

40-43 Address field of a variable-length data

51 51 The address of the field which stored the variable-length data

60 61 Variable-length-data field

70-77 Flag bit

130 CPU

131 Program Memory

132 RAM

133 Input Section

135 Hard Disk (HDD: Record Medium)

137 Display

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号

特開平8-202529

(43) 公開日 平成8年(1996)8月9日

| | | | | |
|---------------------------|---------|---------|---------------|---------|
| (51) Int.Cl. ⁶ | 識別記号 | 庁内整理番号 | F I | 技術表示箇所 |
| G 0 6 F 7/00 | | | | |
| 12/04 | 5 1 0 D | 8323-5E | G 0 6 F 7/ 00 | 1 0 1 V |

審査請求 未請求 請求項の数 7 O L (全 8 頁)

(21) 出願番号 特願平7-8059

(22) 出願日 平成7年(1995)1月23日

(71) 出願人 000001007

キヤノン株式会社

東京都大田区下丸子3丁目30番2号

(72) 発明者 原 廣志

東京都大田区下丸子3丁目30番2号 キヤノン株式会社内

(72) 発明者 為我井 正博

東京都大田区下丸子3丁目30番2号 キヤノン株式会社内

(74) 代理人 弁理士 大塚 康徳 (外1名)

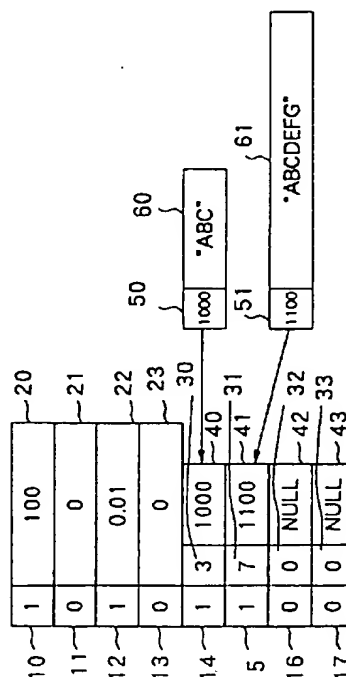
(54) 【発明の名称】 データ処理方法及びその装置

(57) 【要約】

【目的】 固定長データと可変長データとを混在して容易に管理できるデータ処理方法及びその装置を提供することを目的とする。

【構成】 固定長データと可変長データとを混在して処理できるデータ処理装置であって、固定長データ及び可変長データのそれぞれに対応して設けられたフラグビット10～17と、これらフラグビットのそれぞれに対応付けて設けられた固定長データ記録領域20～23と、これらフラグビットのそれぞれに対応付けて設けられた可変長データのデータ長記憶領域30～33、可変長データ記憶領域60、61、及びこれら可変長データ記憶領域のアドレス記憶領域40～43により 固定長データと可変長データとを記憶し、フラグビットが有効である固定長データ及び可変長データを記録媒体に転送して記録する

第4図



【特許請求の範囲】

【請求項 1】 固定長データと可変長データとを混在して処理できるデータ処理装置であって、固定長データ及び可変長データのそれぞれに対応して設けられたフラグと、前記フラグに対応付けて設けられた固定長データ記録領域と、前記フラグに対応付けて設けられた可変長データ記憶領域とを有する記憶手段と、前記記憶手段に記憶されたフラグ、及び前記フラグが有効である固定長データ及び可変長データを記録媒体に転送して記録する記録手段と、を有することを特徴とするデータ処理装置。

【請求項 2】 前記可変長データ記憶領域は、少なくとも当該可変長データのデータ長及び前記可変長データを記憶しているメモリのアドレスを記憶することを特徴とする請求項 1 に記載のデータ処理装置。

【請求項 3】 前記フラグは、少なくとも対応する固定長データ又は可変長データが有効か否かを示す情報、及び対応するデータの種別を示す情報を含むことを特徴とする請求項 1 に記載のデータ処理装置。

【請求項 4】 固定長データと可変長データとを混在して処理するデータ処理方法であって、固定長データを入力すると、対応するフラグを有効にしてメモリに記憶する工程と、可変長データを入力すると、対応するフラグを有効にして当該可変長データのデータ長、及びその格納先アドレスとともに前記可変長データをメモリに記憶する工程と、を有することを特徴とするデータ処理方法。

【請求項 5】 有効なフラグに対応する固定長データ及び可変長データを記録媒体に転送して記録する工程を更に有することを特徴とする請求項 4 に記載のデータ処理方法。

【請求項 6】 前記フラグは、少なくとも対応する固定長データ又は可変長データが有効か否かを示す情報、及び対応するデータの種別を示す情報を含むことを特徴とする請求項 4 に記載のデータ処理方法。

【請求項 7】 前記フラグは更に、対応する固定長データのデータ長を示すデータ長情報を含み、前記長情報に対応して前記固定長データの記憶領域を可変することを特徴とする請求項 4 に記載のデータ処理方法。

【発明の詳細な説明】

【0001】

【産業上の利用分野】 本発明は、固定長データと可変長データとを混在して処理するデータ処理方法及びその装置に関するものである。

【0002】

【従来の技術】 従来、コンピュータで扱われるデータには、8ビット長のデータ、16ビット長のデータ、32ビット長のデータ等があり、また、これらのデータを扱う場合、各データを1つの変数として扱う方法、複数のデータをまとめて1つのブロックとして扱う方法があ

る。

【0003】 図9は、複数のデータを1つのブロックとして扱う場合のデータ構造の例を示す図で、固定長データとして8ビット長の符号付きデータC1、8ビット長の符号無しデータC2、16ビット長の符号無しデータI1、文字列を1つのブロックとして扱う可変長データS1のそれぞれが示されている。このデータブロック内のデータにアクセスする場合は、データブロック全体に対して名前を付与し（図9の場合にはA）、ブロック名と、各データに付加された名前とに基づいて行なわれる。具体例として、図9の“C1”で示されたデータにアクセスする場合は、“A.C1”あるいは“A→C1”といった記号を用いて行う。また図9におけるブロック中、符号無し整数型データを符号付整数型データに変更したデータブロックが同時に必要な場合には、新たなブロックを作成し、そのブロックに別のブロック名（例えばB）を付与して管理する。

【0004】 従って上記従来例において、多数のデータブロックを同時に扱う必要がある場合には、必要な数のデータブロックを作成し、各ブロックにそれぞれ異なる名称を付与して管理しなければならない。よって、プログラムの変更等によって、データブロック内のあるデータを変更する必要がある場合には、その変更が加えられるべき全てのデータブロックに対してデータを変更する必要が生ずる。具体的には、例えば図10の“A”、“B”、“C”で示すデータブロックのそれぞれに新たに32ビット長の符号無しデータを加える場合には、ブロック“A”、“B”、“C”の3つのデータブロックのそれぞれに新たなデータを付加する必要がある。従って、多数のデータブロックに対して、そのデータブロック内のデータの変更或は追加などの必要が生じた場合には、そのデータの変更処理が複雑になる。また、多数のデータブロックに対して変更を行う場合には、誤って変更するなどの可能性もある。このため、扱う固定長データの中でデータ長が最長のデータを十分な数だけ用意し、かつ可変長データに対しても、扱うデータ数に対して十分な数のデータを用意したデータ構造を用い、ブロック内のデータ変更時における処理を簡単にする方法も考えられる。

【0005】 図11はこのような例を示す図で、データブロック120の100～109は固定長データ領域を示している。ここでは固定長データ数は10個である。データブロック121における110～119は可変長データ領域を示し、これら可変長データの数も10個である。この例では、各データブロックのデータ領域へのアクセスは「100+k (0≤k≤9)」あるいは「110+k (0≤k≤9)」で表わされるデータ番号に基づいて行なわれ、また複数のデータブロックが必要な場合には、図11に示すデータブロック120或は121を複数作成することにより対応でき、データ管理、デー

タへのアクセスが共に簡単に行える。

【0006】

【発明が解決しようとする課題】しかしながら、図11に示すデータ構造を用いる場合には、10個の固定長データ、或は10個の可変長データを単位としてデータを管理するために、データを情報記録媒体へ記録する時に、10個の固定長データ或は10個の可変長データを記録する必要がある。このため、固定長データ数或は可変長データ数がそれぞれ“10”未満の場合に、情報記録媒体への記録効率が低下するという問題がある。また、可変長データの場合には、データの再生時、データの長さが予め判別できないために、データ長を調べながらデータを再生する必要があり、データ再生時の再生効率が低下するという問題がある。

【0007】本発明は上記従来例に鑑みてなされたもので、固定長データと可変長データとを混在して容易に管理できるデータ処理方法及びその装置を提供することを目的とする。

【0008】また本発明の目的は、固定長データと可変長データとが混在しているデータブロックにおいて、各データの有効/無効を容易に判断して、効率良く処理できるデータ処理方法及びその装置を提供することにある。

【0009】また本発明の他の目的は、メモリに展開したデータを記録媒体に効率良く記録するデータ処理方法及びその装置を提供することにある。

【0010】

【課題を解決するための手段】上記目的を達成するために本発明のデータ処理装置は以下のような構成を備える。即ち、固定長データと可変長データとを混在して処理できるデータ処理装置であって、固定長データ及び可変長データのそれぞれに対応して設けられたフラグと、前記フラグに対応付けて設けられた固定長データ記録領域と、前記フラグに対応付けて設けられた可変長データ記憶領域とを有する記憶手段と、前記記憶手段に記憶されたフラグ、及び前記フラグが有効である固定長データ及び可変長データを記録媒体に転送して記録する記録手段とを有する。

【0011】上記目的を達成するために本発明のデータ処理方法は以下のような構成を備える。即ち、固定長データと可変長データとを混在して処理するデータ処理方法であって、固定長データを入力すると、対応するフラグを有効にしてメモリに記憶する工程と、可変長データを入力すると、対応するフラグを有効にして当該可変長データのデータ長、及びその格納先アドレスとともに前記可変長データをメモリに記憶する工程とを有する。

【0012】

【作用】以上の構成において、固定長データを入力すると、対応するフラグを有効にしてメモリに記憶し、また可変長データを入力すると、対応するフラグを有効にし

て、その可変長データのデータ長、及びその格納先アドレスとともに、その可変長データをメモリに記憶するように動作する。

【0013】

【実施例】以下、添付図面を参照して本発明の好適な実施例を詳細に説明する。

【0014】〔第1実施例〕図1は本実施例のデータ処理装置の概略構成を示すブロック図である。

【0015】図1において、130はCPUで、プログラムメモリ131に記憶されている制御プログラムに従って装置全体を制御している。132はRAMで、CPU130による処理の実行時、ワークエリアとして使用され、各種データを一時的に保存している。尚、このRAM132は、後述するデータブロックを記憶するのにも使用される。133は、例えばキーボードやポインティングデバイス等の入力部で、オペレータにより操作されて各種データを入力するのに使用される。134はハードディスクコントローラ(HDC)で、ハードディスク・ドライブ(HDD)へのデータの書き込み、及び読出しを制御している。137は液晶やCRT等の表示部で、表示コントローラ(CRTC)の制御の下にビデオメモリ(VRAM)138に記憶されているイメージデータを表示している。139はシステムバスで、前述した各部とCPU130とを接続している。

【0016】図2は、本実施例のデータ処理装置のハードディスク135に記憶されたデータブロックを示す概念図である。また図3は、本実施例のデータ処理装置のRAM132におけるデータブロックの概念図であり、図4は、図3に示すデータブロックに格納されたデータの一例を示す図である。

【0017】図2～図4において、10～13は、対応するデータ領域20～23(固定長データ)にデータが格納されたか否かを示すフラグビット(1ビット)、14～17は、対応する可変長データ領域60、61に可変長データが記憶されているか否かを示すフラグビット(1ビット)である。20～23は固定長データを格納するデータ領域、30～33は可変長データのデータ長を格納するデータ長領域(ここでは8ビット)、40～43は可変長データが格納されたメモリのアドレスを記憶している領域をそれぞれ示している。また50、51は可変長データが格納されているメモリのアドレスを示し、60、61は、メモリに記憶された可変長データそのものを示している。

【0018】本実施例では説明を簡略化するために、このデータブロックに記憶される固定長データ、可変長データのデータ数を共に“4”個とし、可変長データのデータ長を示す領域30～33のそれぞれを8ビット長としている。また、扱う固定長データの種類の8ビット長、16ビット長、32ビット長の3種類とし、ここでは固定長データのデータ長を32ビットとして説明す

【００１９】本実施例のデータブロックにデータを格納する場合は、まず図３に示すデータブロックの、フラグビット１０～１７の全てに“０”を格納して、いずれにもデータが格納されていない状態とする。また、固定長データを格納するデータ領域２０～２３と、データ長領域３０～３３に“０”を格納し、また可変長データのアドレス記憶領域４０～４３に０番地を表す“NULLコード”を格納して、これらデータブロックを初期化する。

10 【0026】このフローチャートで示された処理は、現

【0021】CPU130により実行されているアプリケーションプログラムにより、8ビット長のデータとして、固定長データ領域20にデータ“100”を格納する場合、まずフラグビット10に“1”を格納することによって、固定長データ領域20にデータが格納された状態とする。次にデータ領域20に、8ビット長のデータを32ビット長のデータに変換した“100”を格納する。これには8ビットの2進数“01100100”で表わされるデータを32ビットデータに変換したデータ“000000000000000000000000000000000001100100”を格納する。これと同様に、固定長データ領域22に、32ビット長のデータ“0.01”を格納する場合は、まず固定長データ領域22に対応するフラグビット12に“1”をセットした後、データ領域22にデータ“0.01”を表す32ビット長のデータを格納する。

【００２２】以上の操作により、図４に示すように、固定長データ領域２０、２２のそれぞれに、データ“１００”と“０、０１”とが格納される。

【 0 0 2 3 】次に可変長データ“ABC”を格納する場合について説明する。可変長データ領域60に可変長データ“ABC”を格納する場合、まずフラグビット14に“1”を格納することによって、アドレス記憶領域40にデータが格納されている状態とし、可変長データ長領域30に可変長データ“ABC”の長さ“3”を8ビットデータで格納する。次に、可変長データ“ABC”を実際に記憶する領域60のメモリアドレスを決定し、その可変長データ領域60に可変長データ“ABC”を格納するとともに、そのデータ領域60の先頭アドレス50のアドレス（ここでは“1000”番地）を、アドレス記憶領域40に格納する。これと同様に、可変長データ“ABCDEFGH”を格納する場合には、まず対応するフラグビット15に“1”を格納し、データ長を示す領域31に、この可変長データ“ABCDEFGH”の長さ“7”を格納し、最後に実際に、この可変長データ“ABCDEFGH”が格納される可変長データ領域61の先頭アドレス“1100”番地を、アドレス記憶領域41に格納する。

【 0 0 2 4 ） 以上の操作により、アドレス記憶領域 4 0 , 4 1 で示される各メモリアドレスより、可変長データ“ABC”、“ABCDEFGG”のそれぞれがRAM 1 3 2 に格納される。

【００２５】図５は本実施例のデータ処理装置におけるＲＡＭ１３２にデータブロックを記憶する処理を示すフローチャートで、この処理を実行する制御プログラムはプログラムメモリ１３１に記憶され、ＣＰＵ１３０の制御の下に実行される。

10 【0026】このフローチャートで示された処理は、現在実行中のアプリケーションプログラム或は入力部133よりの指示によりRAM132へのデータの書込みが指示されることにより開始され、まずステップS1でRAM132のフラグビット及びデータ領域に“0”を書込んでクリアする。次にステップS2に進み、書込むべきデータが固定長データか、可変長データかをみる。可変長データであればステップS3に進み、RAM132の対応するフラグビットに“1”をセットし、ステップS4で、それに続くデータ長記憶領域に、その記憶しようとする可変データ長を記憶する。次にステップS5に進み、その可変長データを記憶するRAM132のエリアを探し、そのエリアにその可変長データを記憶するとともに、そのエリアの先頭アドレスをアドレス記憶領域30～33の対応する領域に記憶する。

【0027】一方、ステップS2で固定長データを記憶する場合はステップS7に進み、前述のステップS3と同様に、対応するフラグビットに“1”をセットし、次にステップS8に進んで、そのフラグビットに対応する固定長データ領域に、その固定長データを記憶する。こうして固定長データ或は可変長データがRAM132に記憶されるとステップS6に進み、全てのデータをRAM132に記憶したか否かを判断し、全て記憶すると処理を終了するが、そうでないときはステップS2に戻り、前述の処理を実行する。

【0028】次に、このような処理によってRAM132に格納された固定長データ及び可変長データを、例えばハードディスク135等の情報記録媒体に記録する際の処理を説明する。

【0029】のようにしてRAM132に記憶された
40 データをHDD135に記録する際は、まず、図4に示すフラグビット10～17を図2に示すように記録し、このフラグビットに続けて固定長データ20、22を記録する。更に、可変長データ“ABC”のデータ長“3”30と、その可変長データ“ABC”とを1組とした情報と、可変長データ“ABCDEFGH”のデータ長“7”と、その可変長データ“ABCDEFGH”とを続けて記録する。従って図2に示すように、フラグビット10～17の内、フラグ10、12、14、15に“1”が記録され、これに続いて固定長データ“10
50 0”、“0、01”が記録され、更に可変長データの長

さと可変長データの組である、“3”、“ABC”、“7”、“ABCDEFG”が記録される。

【0030】図6は、RAM132に記憶されたデータをHDD135に記録する処理を示すフローチャートで、この処理を実行する制御プログラムはプログラムメモリ131に記憶されている。

【0031】HDD135への記録が指示されるとステップS10に進み、まずフラグビットの全てをHDD135に記録する。次にステップS11に進み、“1”のフラグビットに対応する固定長データを順次HDD135に記録する。次にステップS12に進み、“1”であるフラグビットに対応する可変長データのデータ長と、その可変長データそのものとを組にしてHDD135に記憶する。

【0032】上記実施例の操作により、複数で、かつ大きさの異なる固定長、可変長のデータを単一のデータ構造で処理することができる簡便なデータ構造を提供でき、またデータブロックの複数のデータの中から、データが格納されているブロックのみを記録媒体に記録することにより、記録媒体へのデータの記録効率を上げることができる。

【0033】【第2実施例】前述の第1実施例では、RAM132のデータブロックにデータを格納する前に、データブロックの初期化としてフラグビット領域に対して“0”を格納すると共に、図3に示す固定長データ領域20～23、データ長記憶領域30～33、アドレス記憶領域40～43にも初期値を格納しているが、フラグビット10～17の値を基に、対応するデータの有無を確認できるため、フラグビット10～17のみに初期値を格納するだけで初期化処理を終了することができる。こうすることにより、データ格納領域に対する初期化処理が省略でき、処理時間の短縮を図ることができる。

【0034】【第3実施例】第1実施例において、固定長データ領域20～23をHDD135に記録する場合、図2に示すように、固定長データ領域20～23に対して固定長の領域を割り当てて記録を行っている。この場合、CPU130により利用されるデータ長、例えば8ビット長、16ビット長、32ビット長データに対して、一定の大きさの領域を常に割り当てなければならない。このため、例えば固定長データのデータ長が8ビットであっても、常に固定長データの最大データ長である32ビット長の記録領域が割り当てられることになり、記録効率を上げるための妨げとなっている。また第1実施例では、フラグビットに対応するデータがどのような種類のものであるかが予め判別できなければ、図2に示すデータを再生する場合にデータ間の区切りを判別することができない。

【0035】そこで、第1実施例に示すフラグビットを複数ビットで構成し、そのビットにデータ種別を表す情

報を付加するように構成することにより、上記問題点を解決することができる。

【0036】図7は、この拡張したフラグビット700を用いて表現できる情報の例を示す図で、図8は第1実施例で示したデータブロックのデータを、図7に示すフラグビットを用いてHDD135へ記録する場合の記録状態の概念図を示す。

【0037】図7に示すように第3実施例では、フラグビット700を3ビットで構成し、8種類のデータを表現できるようにしている。

【0038】また図8において、第1実施例で示した固定長データ“100”、“0.01”を符号無し of データとして扱っている。

【0039】図7に示すフラグビット700により、図8に示すように、領域70～77にデータの種類の表現できるフラグビットが記録され、固定長データ領域20、22では、各データの種類の、そのデータ長がフラグビット70、71より判別可能となっている。即ち、図8の例では、フラグビット70(101)より、固定長データ“100”はデータ長が8ビットで符号無しであり、フラグビット72(111)より、固定長データ“0.01”はデータ長が32ビットで符号無しであることが分かる。更に、フラグビット74、75(100)により、可変長データは文字列であることが分かる。

【0040】こうして、各固定長データに対して必要なデータ領域のみを確保することが可能となり、前述の第1実施例よりも、HDD135(記録媒体)へのデータの記録効率をさらに向上させることができる。また、記録されているデータの種類の、このフラグビットから容易に判別できるようになる。

【0041】尚、本発明は、複数の機器から構成されるシステムに適用しても、1つの機器から成る装置に適用しても良い。また、本発明はシステム或は装置に本発明を実施するプログラムを供給することによって達成される場合にも適用できる。

【0042】以上説明したように本実施例によれば、データ長がM以下の、複数のデータ長が異なる固定長データ、及び最大K個の可変長データから構成される複数のデータブロックを扱うデータ構造において、各データブロックをデータ長Mの複数の固定長データと、K個の可変長データとを有し、かつ各データブロック中の個々のデータに対してデータの有効或は無効を示すフラグビットを有するデータブロックで表現することにより、データブロックの構造を簡略化できる。

【0043】また、データブロックを記録媒体へ記録する際に、フラグビットの内容から有効なデータのみを記録媒体に記録し、無効なデータを記録しない様に構成することによって、記録媒体への記録効率を向上できる効果がある。

【0044】

【発明の効果】以上説明したように本発明によれば、固定長データと可変長データとを混在して容易に管理できる効果がある。

【0045】また本発明によれば、固定長データと可変長データとが混在しているデータブロックにおいて、各データの有効／無効を容易に判断して、効率良く処理できる効果がある。

【0046】また本発明によれば、メモリに展開したデータを記録媒体に効率良く記録できる効果がある。

【0047】

【図面の簡単な説明】

【図1】本実施例のデータ処理装置の概略構成を示すブロック図である。

【図2】本実施例のデータ処理装置において、記録媒体に記録されるデータの構成を示す概念図である。

【図3】本実施例のデータ処理装置におけるRAM上のデータ構造を示す概念図である。

【図4】本実施例のデータ処理装置のRAM上におけるデータの記憶状態の具体例を示す図である。

【図5】本実施例のデータ処理装置におけるメモリへのデータ書き込み処理を示すフローチャートである。

【図6】本実施例のデータ処理装置におけるメモリに記憶されたデータをHDDに記録する処理を示すフローチャートである。

10

*【図7】本発明の第3実施例に係るフラグビットの状態を示す図である。

【図8】本発明の第3実施例におけるデータブロックの記録媒体への記録状態を示す概念図である。

【図9】複数のデータをブロックとして扱う場合のデータ構造の従来例を説明する図である。

【図10】複数のデータをブロックとして扱う場合のデータ構造の従来例を説明する図である。

【図11】複数のデータをブロックとして扱う場合のデータ構造の従来例を説明する図である。

【符号の説明】

10～17 フラグビット

20～23 固定長データ領域

30～33 可変長データのデータ長領域

40～43 可変長データのアドレス領域

51, 51 可変長データを格納した領域のアドレス

60, 61 可変長データ領域

70～77 フラグビット

130 CPU

20 131 プログラムメモリ

132 RAM

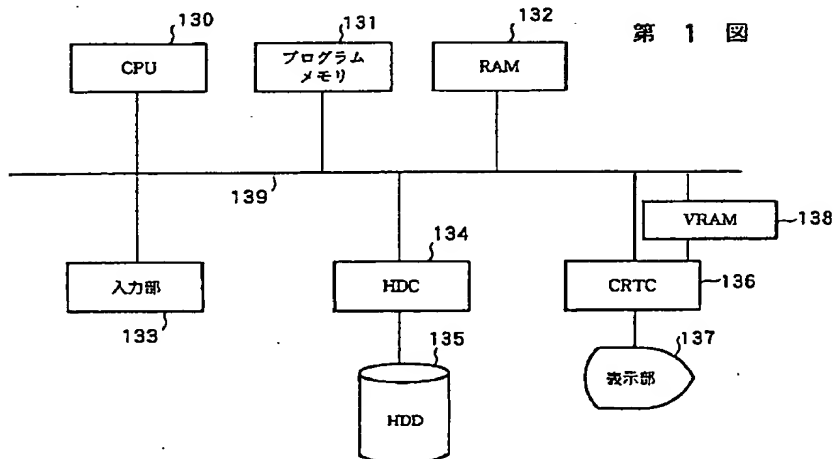
133 入力部

135 ハードディスク（HDD：記録媒体）

137 表示部

*

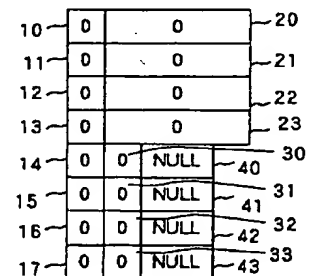
【図1】



第 1 図

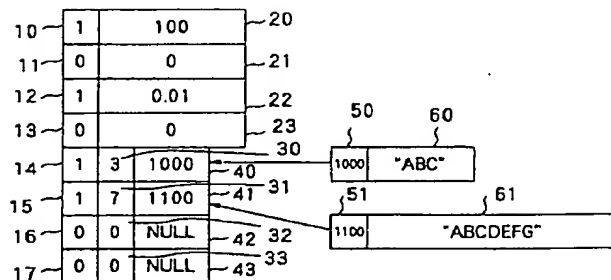
【図3】

第 3 図

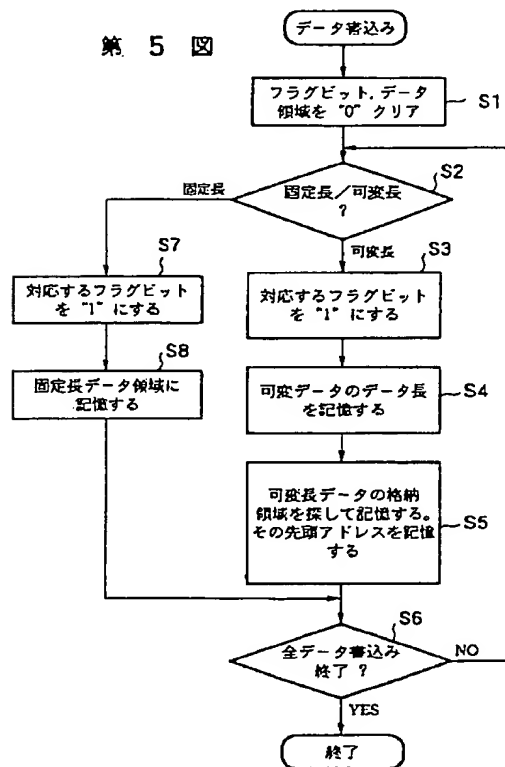


【圖 4】

第 4 圖

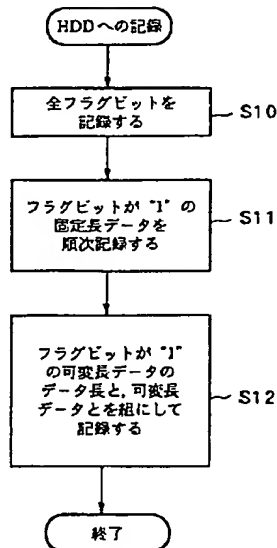


第 5 回



【図7】

第 7 図



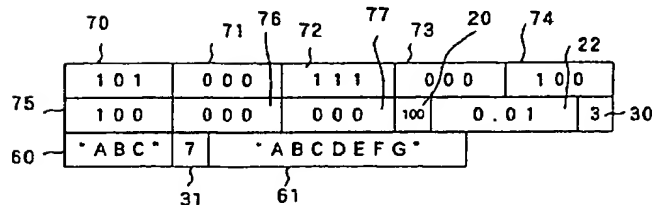
| 700 | |
|-------|-----------------|
| 0 0 0 | データの格納なし |
| 0 0 1 | データ長 8 ビット符号あり |
| 0 1 0 | データ長 16 ビット符号あり |
| 0 1 1 | データ長 32 ビット符号あり |
| 1 0 0 | 可変長変数 (文字列) |
| 1 0 1 | データ長 8 ビット符号なし |
| 1 1 0 | データ長 16 ビット符号なし |
| 1 1 1 | データ長 32 ビット符号なし |

【圖 8】

第 9 圖

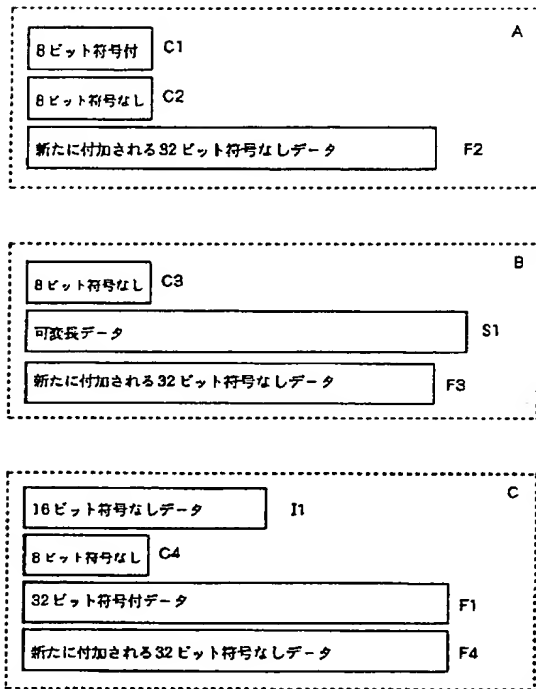
Diagram illustrating a data structure layout with four fields:

- 8ビット符号付 (C1)
- 8ビット符号なし (C2)
- 16ビット符号なし (I1)
- 文字列 (S1)



【図10】

第 10 図



【図11】

第 11 図

